

Rotation Invariant Angular Descriptor Via A Bandlimited Gaussian-like Kernel

Michael T. McCann, Matthew Fickus, Jelena Kovačević

Abstract

We present a new smooth, Gaussian-like kernel that allows the kernel density estimate for an angular distribution to be exactly represented by a finite number of its Fourier series coefficients. Distributions of angular quantities, such as gradients, are a central part of several state-of-the-art image processing algorithms, but these distributions are usually described via histograms and therefore lack rotation invariance due to binning artifacts. Replacing histogramming with kernel density estimation removes these binning artifacts and can provide a finite-dimensional descriptor of the distribution, provided that the kernel is selected to be bandlimited. In this paper, we present a new band-limited kernel that has the added advantage of being Gaussian-like in the angular domain. We then show that it compares favorably to gradient histograms for patch matching, person detection, and texture segmentation.

Keywords: kernel density estimation, angular distribution, patch descriptor, person detection, segmentation

1. Introduction

Histograms of angular quantities are a key component of many of the most successful algorithms for a variety of image processing tasks. For example, SIFT [1], along with some of its variants including GLOH [2], SIFT+GC [3], and CSIFT [4] (but not SURF [5] or PCA-SIFT [6]), use histograms of local gradient angles to form a keypoint descriptor. SIFT descriptors are widely used, with applications including medical image registration [7], human activity analysis [8], and object recognition [9]. HOG [10] and its extensions, such as part-based models [11], calculate local gradient histograms at every point in an image and are useful in human [12] and object [13] detection.

Despite their widespread use in vision, histograms have a fundamental weakness when estimating distributions of angular quantities because they rely on binning and are therefore not invariant to rotation. That is, a rotation of the input angles results in a rotation of the histogram plus distortion; see Figure 1. This problem affects even methods that attempt to be invariant to rotation. For example, SIFT [1] builds histograms with respect to a dominant angle, but this dominant angle is itself estimated from an angular histogram. The radial

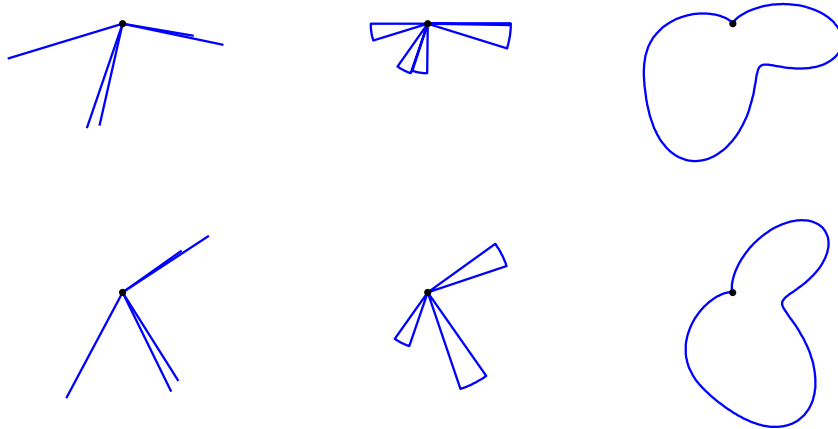


Figure 1: Top row: A weighted set of angles, its 20-bin histogram, and the estimate based on the proposed method (FS-KDE) using 20 numbers. Bottom row: The same set of angles rotated counter-clockwise by 45° , its histogram, and FS-KDE. While the rotation distorts the histogram, it only causes a corresponding rotation in the FS-KDE.

gradients computed in RIFF [14] are invariant to rotation, but are collected into angular histograms which are not.

A rotation-invariant alternative to the histogram is the kernel density estimate (KDE) [15], which estimates a continuous distribution from its samples by putting a lump (kernel) of density at the location of each sample. KDEs create smooth estimates and, under certain assumptions, converge to the correct distribution with fewer samples than histograms [15]; however, KDEs are not useful as descriptors in image processing, because evaluating the KDE at a point requires all of the samples to be stored in memory and there is no straightforward way to compute distances between KDEs. A variant of the KDE that helps to address these limitations is characteristic function estimation [16], wherein the characteristic function (or, in the language of signal processing, Fourier series) of a distribution is estimated, rather than its angular-domain version. For angular distributions, the estimated Fourier series is discrete and can be truncated to form a finite-length descriptor. Reference [17] explores the use this type of descriptor as a replacement for histograms inside of HOG [10]. The problem with this truncation is that it is equivalent to convolving the angular kernel of a KDE with a sinc function. These kernels may then have undesirable angular domain properties, such as attaining negative values or being non-monotonic on the intervals $[-\pi, 0]$ and $[0, \pi]$. For example, since [17] uses a Dirac kernel in the angular domain and then truncates the Fourier series, the effective angular kernel is a sinc function.

In this work, we present a new kernel designed to have good properties in

the angular domain while simultaneously being band-limited in the frequency domain, meaning that its Fourier series has a finite number of non-zero terms and can therefore be used directly as a descriptor without truncation.

2. Fourier Series Kernel Density Estimation

We call the method of representing an angular KDE via its Fourier series the *Fourier Series-Kernel Density Estimate* (FS-KDE). In this section, we develop our notation for the FS-KDE and describe its properties.

2.1. Definition of the FS-KDE

Given an angle-weight pair, (Θ, W) , consisting of a set of angles, $\Theta = \{\theta_0, \theta_1, \dots, \theta_{N-1}\}$, and a set of positive scalar weights, $W = \{w_0, w_1, \dots, w_{N-1}\}$, we form a KDE, $f : [-\pi, \pi] \rightarrow \mathbb{R}$, of their underlying distribution as a sum of kernels,

$$f(\theta) = \frac{1}{N} \sum_{n=0}^{N-1} w_n h(\theta - \theta_n),$$

where the kernel, $h(\theta)$, is a positive function that integrates to one.¹ For example, the angle-weight pair might come from the angles and magnitudes of the gradients in an image.

We can then expand h in terms of its Fourier series and rearrange terms,

$$\begin{aligned} f(\theta) &= \frac{1}{N} \sum_{n=0}^{N-1} w_n \sum_{k=-\infty}^{\infty} H_k e^{jk(\theta - \theta_n)} \\ &\stackrel{(a)}{=} \frac{1}{N} \sum_{n=0}^{N-1} w_n \sum_{k=-K}^K H_k e^{jk(\theta - \theta_n)} \\ &= \sum_{k=-K}^K \underbrace{\left(\frac{H_k}{N} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n} \right)}_{F_k} e^{jk\theta}, \end{aligned} \tag{1}$$

where (a) holds for bandlimited kernels. Equation (1) is the expression of f in terms of its Fourier series coefficients, F_k . We denote the relationship between $f(\theta)$ and F_k as $f(\theta) \xleftrightarrow{FS} F_k$. From (1), we see that f is bandlimited: it has $2K + 1$ non-zero Fourier series coefficients. We also see that F_{-k} is the complex conjugate of F_k , so, in practice, only $K + 1$ complex values must be computed and stored to represent f . Thus, an FS-KDE of order K takes the same amount of storage as a histogram with $2(K + 1)$ bins.

¹For greater flexibility, we do not require f to integrate to one; we thus use the term *distribution* loosely.

2.2. Properties

We now discuss some useful properties of the FS-KDE. First, since F is simply the Fourier series representation of f , we can leverage all of the properties of the Fourier series [18]. Of specific interest here are linearity, $\alpha f(\theta) + \beta g(\theta) \xrightarrow{FS} \alpha F_k + \beta G_k$, and Parseval's equality, $\|f\|^2 = 2\pi\|F\|^2$. Together, these mean that the distance between two FS-KDEs, $\|f - g\|^2$, can be computed as the finite sum $2\pi\|F - G\|^2$.

The FS-KDE is rotation invariant in the sense that a rotation of the angles in the angle-weight pair results in a corresponding rotation in the FS-KDE. To be more precise, begin with an angle-weight pair (Θ, W) . Form its rotation, (Θ_ϕ, W) , where $\Theta_\phi = \{\theta_0 + \phi, \theta_1 + \phi, \dots, \theta_{N-1} + \phi\}$. If F is the FS-KDE for (Θ, W) and F_ϕ is the FS-KDE for (Θ_ϕ, W) , then from (1),

$$\begin{aligned} F_{\phi,k} &= \frac{H_k}{N} \sum_{n=0}^{N-1} w_n e^{-jk(\theta_n + \phi)} \\ &= e^{-jk\phi} \frac{H_k}{N} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n} \\ &= e^{-jk\phi} F_K. \end{aligned} \tag{2}$$

By the shift in time property of Fourier series, (2) means that f_ϕ is equal to f circularly shifted by ϕ . Thus, a rotation in the input angles has caused a corresponding rotation in the KDE.

2.3. FS-KDEs for Images

Several computer vision algorithms estimate local angular distributions (usually via histograms) for every location in an image. For example, this is the approach of deformable parts models [11] for object detection. In this section, we describe efficient computation of local FS-KDE estimates on images via linear filtering.

Let $(\Theta(x), W(x))$ be a weighted angular image, where $\Theta : X \rightarrow [-\pi, \pi]$ is an image of angles and $W : X \rightarrow \mathbb{R}$ is a corresponding image of weights, where X is a discrete set of pixel locations (e.g. $\mathbb{Z}_{1200} \times \mathbb{Z}_{1600}$). For example, $(\Theta(x), W(x))$ may be formed from computing the gradient of an intensity image. (Note that we write the argument x in $(\Theta(x), W(x))$ to make a distinction between the weighted angular image and the angle-weight pair we introduced in Section 2.1.)

We aim to compute a KDE around each point $x \in X$, with a neighborhood defined by φ , a positive window function with $\|\varphi\|_1 = 1$. We define the FS-KDE for $(\Theta(x), W(x))$ at location x and angle θ as

$$f(x, \theta) = \sum_{y \in X} W(y) h(\theta - \Theta(y)) \varphi(x - y),$$

Then, following a similar procedure as in Section 2.1, we arrive at an expression for $f(x, \theta)$ in terms of its Fourier series coefficients,

$$f(x, \theta) = \sum_{k=-K}^K F_k(x) e^{jk\theta},$$

where $F_k(x) = H_k(W(\cdot)e^{-jk\Theta(\cdot)} * \varphi)(x)$, the symbol $*$ denotes discrete time convolution, and $W(\cdot)e^{-jk\Theta(\cdot)}$ is computed pointwise. This means that local FS-KDEs of order K can be computed via $K + 1$ complex filtering operations.

3. Bandlimited Gaussian-like Kernel

In this section we present a new Gaussian-like kernel for use in FS-KDEs, which we call the \cos^{2K} kernel.

3.1. Kernel Selection

Any kernel that has a bandlimited Fourier series can be used to form an FS-KDE using (1), however we argue for the following additional requirements to make the kernel reasonable for estimation of angular distributions: (1) the kernel should be real; (2) the kernel should be non-negative; (3) the kernel should be an even function; (4) the kernel should integrate to one; and (5) the kernel should take the value zero at π .

We propose the \cos^{2K} kernel,

$$h(\theta) = C_K \cos^{2K} \left(\frac{\theta}{2} \right), \quad (3)$$

where C_K is a normalizing constant and K , which we call the *order*, controls the width of the kernel (Figure 2). The \cos^{2K} kernel clearly satisfies requirements 1-5 above. We will now show that it is also bandlimited and Gaussian-like.

Bandlimited. We can rearrange (3) to reveal its Fourier series coefficients,

$$\begin{aligned} h(\theta) &\stackrel{(a)}{=} C_K \left(\frac{e^{\frac{j}{2}\theta} + e^{-\frac{j}{2}\theta}}{2} \right)^{2K} \\ &\stackrel{(b)}{=} \frac{C_K}{2^{2K}} \sum_{p=0}^{2K} \binom{2K}{p} e^{\frac{j}{2}p\theta} e^{-\frac{j}{2}(2K-p)\theta} \\ &\stackrel{(c)}{=} \sum_{p=0}^{2K} \frac{C_K}{2^{2K}} \binom{2K}{p} e^{j(p-K)\theta} \\ &\stackrel{(d)}{=} \sum_{k=-K}^K \underbrace{\left(\frac{C_K}{2^{2K}} \binom{2K}{K+k} \right)}_{H_k} e^{jk\theta} \end{aligned} \quad (4)$$

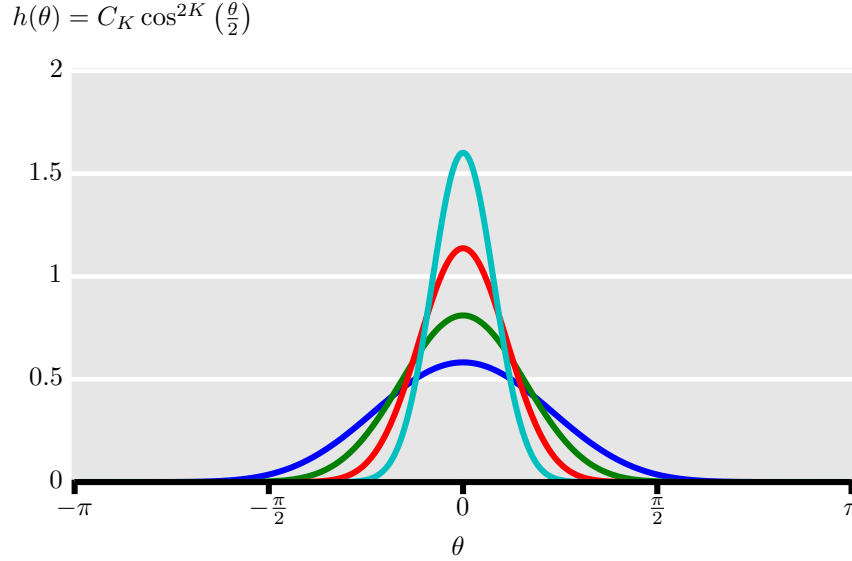


Figure 2: Examples of the proposed \cos^{2K} kernel for $K = 4, 8, 16$, and 32 , with $K = 32$ being the narrowest of the kernels shown. Each kernel is 2π -periodic and integrates to one. As K increases, the kernels become sharper.

where (a) follows from Euler’s formula; (b) from the binomial theorem; (c) from an interchange of finite sums and combining the exponents; and (d) from the substitution $k = p - K$. This expression for $h(\theta)$ confirms that it is bandlimited.

Gaussian-like. The Gaussian kernel,

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}.$$

is ubiquitous, but does not work as an FS-KDE kernel because it is neither bandlimited nor defined on a circular domain.² We will show that the \cos^{2K} is Gaussian-like in that its derivatives behave in a similar way. The derivatives of the Gaussian are

$$g^{(n)}(x) = \frac{d^n}{(dx)^n} g(x) = (-1)^n He_n\left(\frac{x}{\sigma}\right) g(x),$$

where $He_n(x)$ is the n th order Hermite polynomial in x . This is useful because we know that $He_n(x)$ has n real roots, each with multiplicity one. Therefore, $g^{(n)}(x) = 0$ for n values of x (and also tends to zero as $|x|$ tends to infinity).

²There are several adaptations of the Gaussian to a circular domain, including the von Mises and circularly extended Gaussian, neither of which are bandlimited and so are not suitable FS-KDE kernels.

We want to show that the \cos^{2K} kernel (3) is Gaussian-like, i.e. $h^{(n)}(x) = 0$ for n values of $\theta \neq \pi$ (our logic for ignore zeros at π is by the analogy $\lim_{x \rightarrow \infty} g(x) \sim h(\pi)$). By the definition of the Fourier series and the fact that $h(\theta)$ is bandlimited, we have

$$h(\theta) = \sum_{k=-K}^K H_k e^{jk\theta}.$$

We know that the H_k are real and $H_k = H_{-k}$ because $h(\theta)$ is even and real. Thus h is just a polynomial of degree $2K$, $h(\theta) = e^{-jK\theta} P(e^{jk\theta})$. By the fundamental theorem of algebra, the number of roots of $P(z)$ is $2K$; if we call those roots $z_0, z_1, \dots, z_{2K-1}$, then the zeros of h correspond to the unit-modulus z_i : $h(\arg(z_i)) = 0$ for each z_i with $|z_i| = 1$, where we use $\arg(z)$ to denote the argument of the complex number z .

The derivatives of $h(\theta)$ are

$$h^{(n)}(\theta) = \frac{d^n}{(d\theta)^n} h(\theta) = \sum_{k=-K}^K (jk)^n H_k e^{jk\theta},$$

These derivatives are also polynomials of degree $2K$, which we call $P^{(n)}$, with the same relationship between the roots of the polynomial and the zeros of $h^{(n)}$ as for h and P .

We see from (3) that all of the zeros of h are at $\theta = \pi$, meaning that $P(z)$ has a root at $z = -1$ with multiplicity $2K$. This also means that each $P^{(n)}$ has a root at $z = -1$ with multiplicity $2K - n$ (by the chain rule). Thus the number of zeroes of $h^{(n)}(\theta)$ for $\theta \neq \pi$ is less than or equal to n .

On the other hand, because $h(-\pi) = h(\pi) = 0$, the mean value theorem guarantees that there exists a $\theta_0 \in (-\pi, \pi)$ such that $h^{(1)}(\theta_0) = 0$. Repeating the same argument gives $\theta_{1,0} \in (-\pi, \theta_0)$ and $\theta_{1,1} \in (\theta_0, \pi)$ such that $h^{(2)}(\theta_{1,0}) = h^{(2)}(\theta_{1,1}) = 0$ and so on for each $h^{(n)}$. Thus the number of zeroes of $h^{(n)}(\theta)$ for $\theta \neq \pi$ is greater than or equal to n .

Combining the inequalities from the previous two paragraphs, we have $h^{(n)}(x) = 0$ for n values of $\theta \neq \pi$.

3.2. Practical Considerations

In this section, we explore a few practical considerations that must be taken into account when computing FS-KDE using our \cos^{2K} kernel, including calculation of C_K , a normal approximation to (1), and creating approximate FS-KDEs via truncation.

3.2.1. Normalization

We have not yet calculated C_K , the normalizing constant for the kernel h in (3). We do this via (4), giving

$$\begin{aligned} \int_0^{2\pi} h(\theta) d\theta &= \int_0^{2\pi} \sum_{k=-K}^K \left(\frac{C_K}{2^{2K}} \binom{2K}{K+k} \right) e^{jk\theta} d\theta \\ &\stackrel{(a)}{=} \int_0^{2\pi} \frac{C_K}{2^{2K}} \binom{2K}{K} d\theta \\ &= \frac{C_K}{2^{2K-1}} \binom{2K}{K} \pi, \end{aligned}$$

where (a) comes from the symmetry of $e^{jk\theta}$ for $k \neq 0$. So to make the \cos^{2K} kernel integrate to one, we set

$$C_K = \frac{2^{2K-1}}{\binom{2K}{K} \pi}. \quad (5)$$

Thus, the formula for Fourier series coefficients of the FS-KDE using the \cos^{2K} kernel is

$$F_k = \frac{(K!)^2}{2\pi N (K-k)!(K+k)!} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n}. \quad (6)$$

3.2.2. Normal Approximation

For large K s, the binomial coefficients in (1) and (5) can be replaced with a normal approximation,

$$\binom{n}{k} \approx \frac{2^n}{\sqrt{n\pi/2}} e^{-\frac{(k-(n/2))^2}{n/2}},$$

giving an approximate version of (6),

$$F_k = \frac{1}{2\pi N} e^{-k^2/K} \sum_{n=0}^{N-1} w_n e^{-jk\theta_n}. \quad (7)$$

This approximation saves computation as compared to (6) and also reveals that the F_k s decay exponentially. The quality of the normal approximation improves as K increases; in our implementation we switch from (6) to (7) when $2K \geq 80$.

3.2.3. Truncation

In our current formulation, the bandwidth of the kernel density estimate is controlled by K , which also governs how many Fourier series terms are nonzero. Careful inspection of Figure 2 reveals that the \cos^{2K} kernel does not sharpen quickly as K increases: a sharp kernel requires a large K and therefore a long descriptor. One way to achieve sharp kernels with a shorter descriptor is through

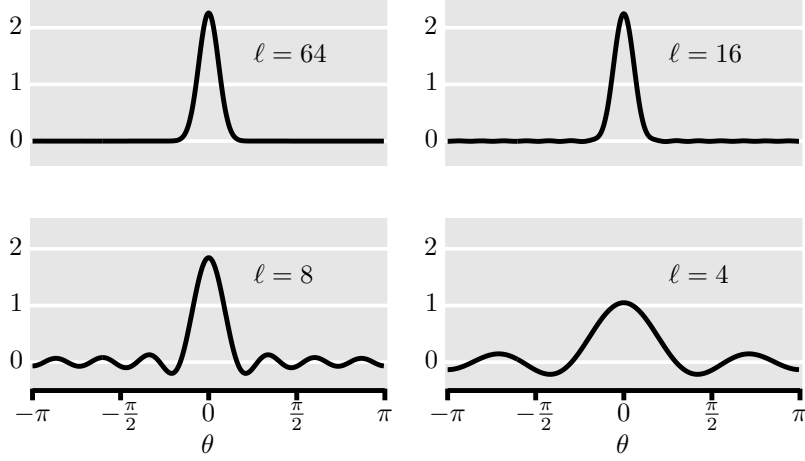


Figure 3: Examples of the \cos^{2K} kernel of order 64 with different levels of truncation, where ℓ is the number of non-zero coefficients. Distortion is barely noticeable even when three quarters of the F_k s are set to zero (upper-right panel).

truncation. The approximation (7) reveals that when K is large, the decaying exponential term will cause $|F_k|$ to be very small for k near K . In fact,

$$\max_{\Theta} \frac{|F_k|}{|F_0|} = e^{-k^2/K}.$$

Thus, for a fixed K and a small truncation threshold ϵ , we create a truncated FS-KDE, \hat{F} , according to

$$\hat{F}_k = \begin{cases} 0 & \text{if } e^{-k^2/K} < \epsilon, \\ F_k & \text{otherwise.} \end{cases} \quad (8)$$

In the angular domain, truncation introduces distortion into the kernel, but this distortion is slight even when many coefficients are truncated (see Figure 3). We provide MATLAB code for the FS-KDE using the \cos^{2K} kernel in the reproducible research compendium for this article, [19].

4. Canonicalization

We showed in Section 2.2 that FS-KDEs are rotation invariant in the sense that a rotation of the input angles causes a corresponding rotation in the density estimate. We may, however, also desire that a rotation of the input angles cause no change at all to the estimated distribution. This would be useful if, e.g., FS-KDEs are being used as point descriptors in an image matching application. We can achieve this by rotating FS-KDEs to a standard, or *canonical*, position,

such that all FS-KDEs that are rotations of each other end up with the same canonical version. In this section, we present two methods of achieving this canonicalization.

4.1. F_1 Canonicalization

A natural way of canonicalizing an angle-weight pair, (Θ, W) , is to rotate the angles such that their mean is equal to zero. One way to define the angular mean is to assign to each angle θ_n a complex number, $z_n = w_n e^{j\theta_n}$ with modulus w_n and argument θ_n , and then sum these numbers and take the argument of the result, $\bar{\theta} = \arg\left(\sum_{n=0}^{N-1} z_n\right)$. Then, the canonical angle-weight pair is $(\tilde{\Theta}, W)$, where $\tilde{\Theta} = \Theta_{-\bar{\theta}}$, which is the rotation of Θ by $-\bar{\theta}$.

From (1) we see that the argument of the first Fourier series coefficient of the corresponding FS-KDE, $\arg(F_1)$, is equal to $-\bar{\theta}$ so long as H_1 is real and positive, as is the case for the \cos^{2K} kernel. As a result, canonicalizing the angle-weight pair causes F_1 to be real because $\bar{\theta} = 0$. Using this fact, we can directly canonicalize an FS-KDE, f , without having to know the angle-weight pair that it came from. We define the canonical version of f as

$$\tilde{f}(\theta) \xleftrightarrow{FS} \tilde{F}_k = e^{-jk \arg(F_1)} F_k, \quad (9)$$

which is the rotation of f that makes F_1 real.

We now show that this canonicalization has the property that rotating a set of angles does not change its canonical FS-KDE. In other words, all rotations of an angle-weight pair have the same canonical FS-KDE, \tilde{f} .

Lemma 1. *Let (Θ, W) be an angle-weight pair, let (Θ_ϕ, W) be its rotation by ϕ , and let F and F_ϕ be their FS-KDEs. Then,*

$$\tilde{F}_{\phi,k} = \tilde{F}_k$$

PROOF. Beginning with an FS-KDE (1), we have

$$\begin{aligned} \arg(F_{\phi,1}) &= \arg\left(\frac{H_1}{N} \sum_{n=0}^{N-1} w_n e^{-j(\theta_n + \phi)}\right) \\ &= \arg\left(e^{-j\phi} \frac{H_1}{N} \sum_{n=0}^{N-1} w_n e^{-j\theta_n}\right) \\ &= \arg(F_1) - \phi. \end{aligned}$$

We know from Section 2.2 that $F_{\phi,k} = e^{-jk\phi} F_k$, and thus using the definition of canonicalization (9),

$$\begin{aligned} \tilde{F}_{\phi,k} &= e^{-jk \arg(F_{\phi,1})} F_{\phi,k} \\ &= e^{-j(\arg(F_1) - \phi)k} e^{-jk\phi} F_k \\ &= e^{-jk \arg(F_1)} F_k = \tilde{F}_k. \end{aligned}$$

4.2. Stability of F_1 Canonicalization

Now that we have shown that F_1 canonicalization aligns distributions that are exact rotations of each other, we study its effect on distributions that are noisy rotations of each other. Intuitively, a good canonicalization will give similar canonical versions to all FS-KDEs that are noisy rotations of each other; We call this property *stability*. Conversely, a bad canonicalization might amplify small amounts of noise, assigning similar FS-KDEs very different canonical versions. The following theorem states that the stability of F_1 canonicalization is related to the magnitude of the first Fourier series coefficient of the distribution that is being canonicalized, $|F_1|$. We leave the proof of the theorem to Appendix A.

Theorem 1 (Stability of F_1 Canonicalization). *Let (Θ, W) be an angle-weight pair. Without loss of generality, assume $\bar{\theta} = 0$ and $\sum_{n=0}^{N-1} w_n = N$. Let $(\Theta^\epsilon, \alpha W^\epsilon)$ be its noisy version such that $w_n^\epsilon e^{j\theta_n^\epsilon} = w_n e^{j\theta_n} + \epsilon_n$ where the ϵ_n are drawn according to the complex normal distribution with mean zero and standard deviation σ/\sqrt{N} (i.e., the imaginary part of $\epsilon_n \sim \mathcal{N}(0, \sigma^2/N)$ and the real part of $\epsilon_n \sim \mathcal{N}(0, \sigma^2/N)$), and α scales W^ϵ such that $\sum_{n=0}^{N-1} \alpha w_n^\epsilon = N$. Let f and f^ϵ be the K th-order \cos^{2K} FS-KDEs of (Θ, W) and $(\Theta^\epsilon, W^\epsilon)$, respectively. Then*

$$\mathbb{E} \left[\|F^\epsilon - \tilde{F}^\epsilon\| \right] \leq \mathbb{E} \left[\sqrt{\sum_{k=-K}^K \left(2B_k N \sin \left(\frac{k}{2} \arctan \left(\frac{B_1 \epsilon}{|F_1| + B_1 v} \right) \right) \right)^2} \right],$$

with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, $v \sim \mathcal{N}(0, \sigma^2)$, and B_k the coefficient from (1), $B_k = \frac{C_K}{2^{2K} N} \binom{2K}{K+k}$.

To make use of Theorem 1, we note that the distance between a canonical distribution and its canonical noisy version is bounded by the distance due to noise and the distance due to canonicalizing the noisy version, i.e. $\|F - \tilde{F}^\epsilon\| \leq \|F - F^\epsilon\| + \|F^\epsilon - \tilde{F}^\epsilon\|$. The theorem lets us calculate the expected value of $\|F^\epsilon - \tilde{F}^\epsilon\|$ only as a function of $|F_1|$ relative to the variance of the noise, σ , without having to know f or f^ϵ . Notably, $\mathbb{E} [\|F^\epsilon - \tilde{F}^\epsilon\|]$ approaches zero as $|F_1|$ grows relative to the noise. Because the norm is always non-negative, its expected value approaching zero implies that its variance is also approaching zero. This means that as noise gets smaller, $\|F - \tilde{F}^\epsilon\| \approx \|F - F^\epsilon\| \approx 0$, which is what we set out to show.

We illustrate this with a simulation (Figures 4 and 5). We first generate two random distributions, one with a large $|F_1|$ and one with a small $|F_1|$. For each of these distributions, we generate noisy versions for a range of noise levels and calculate $\|F - F^\epsilon\|$ and $\|F^\epsilon - \tilde{F}^\epsilon\|$. Comparing Figures 4 and 5, we see that $\|F^\epsilon - \tilde{F}^\epsilon\|$ is expected to be smaller in Figure 4, where $|F_1|$ is large. For comparison, we plot the distance caused by rotating these distributions,

$\|F - F_\phi\|$ (Figure 4(e) and 5(e)). In Figure 4, the distance caused by rotation is larger than the distance due to canonicalizing the noisy versions, but in Figure 5, the distance due to canonicalization is significant compared to the rotation distance.

As a concrete example, take a patch matching application, such as we describe in Section 5.1. If distributions in a dataset are randomly rotated and $|F_1|$ for each patch is large relative to the expected noise, it makes sense to canonicalize the distributions before matching because much of the distance between corresponding patches will come from their rotation, which canonicalization will remove. If, on the other hand, patches in the dataset are not rotated, canonicalization will hurt performance because $\|F^\epsilon - \tilde{F}^\epsilon\|$ will increase the distance between matching patches. As $|F_1|$ for the patches shrinks relative to the noise, canonicalization becomes increasingly unstable. This is because when $|F_1|$ is small, a small amount of noise can greatly affect $\arg(F_1)$. This situation can arise in two ways. The first is when all weights are small, meaning the distribution being calculated is essentially zero; instability is no problem in this case because rotation has no effect on FS-KDEs that are nearly zero. The second is when the distribution has symmetry, e.g., when Θ contains only angles only at zero and π . Such cases may arise in practice, leading us to explore a generalization of F_1 canonicalization that can remove these symmetries.

4.3. F_k Canonicalization

We can generalize the idea in (9) to rotating f by an angle, $(\arg(F_\ell) + 2\pi n)/\ell$, such that F_ℓ is real. The added complexity is that for $\ell \geq 2$, this angle is not unique; it can take ℓ different values. We disambiguate these by defining F_k canonicalization recursively,

$$\tilde{f}^\ell(\theta) \xleftrightarrow{FS} \tilde{F}_k^\ell = e^{-jk \arg(F_\ell)/\ell} \tilde{F}_k^{\ell-1},$$

with $\tilde{F}_k^1 = \tilde{F}_k$ as defined in (9). One way to think about this process is that we first F_1 canonicalize, then we pick the smallest rotation that makes F_2 real, then pick the smallest rotation that makes F_3 real, and so on until F_ℓ . For any choice, $1 \leq \ell \leq K$, we can show that rotating the input set of angles does not affect the canonical version, using the same steps as for F_1 canonicalization.

The benefit of using $\ell > 1$ is that for angular distributions with a certain kind of symmetry, $|F_1|$ may be small (and thus F_1 canonicalization will not be robust to noise), while, e.g., $|F_2|$ may be large, meaning F_2 canonicalization will be robust to noise. The trade-off is that if $|F_1|$ and $|F_2|$ are of similar size, F_1 canonicalization will be more robust to noise. (To see this, note that F_2 canonicalization is just another mean subtraction, except that the mean is calculated by first doubling all the angles in Θ . This doubling can remove unwanted symmetry, but it also amplifies noise.)

In our experiments, we leverage this in the following way: When what is important is pairwise distances between FS-KDEs, then we can define a canonical distance,

$$\|f - g\|_{\text{canonical}} = 2\pi \min_{\phi} \|F - e^{-jk\phi} G\|.$$

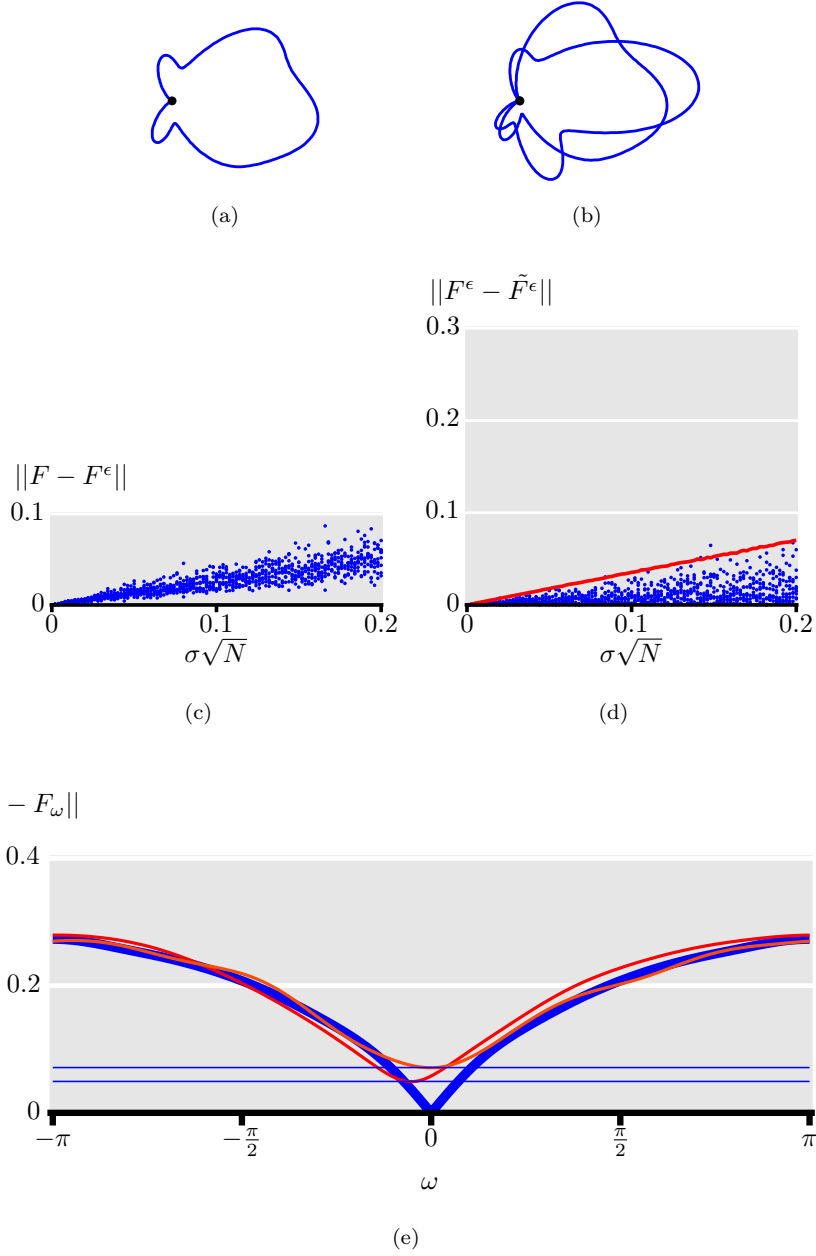


Figure 4: (a) A distribution, f . (b) Two of its noisy versions, f^ϵ . (c) The distance between f and its noisy versions plotted as a function of increasing noise. (d) The additional error introduced by F_1 canonicalization of the noisy versions, along with the expectation from Theorem 1. Because this distribution has a large $|F_1|$ value compared to the noise, F_1 canonicalization makes small changes to noisy versions of f . (e) Curves indicate $\|F - F_\omega\|$ (bold line) and $\|F - F_\phi^\epsilon\|$ (thin lines) for f , two noisy versions of f , and their rotations by ϕ . Because $|F_1|$ is large compared to the noise, $\|F - \tilde{F}_\phi^\epsilon\|$ (horizontal lines) is almost as small as $\min_\phi \|F - F_\phi^\epsilon\|$.

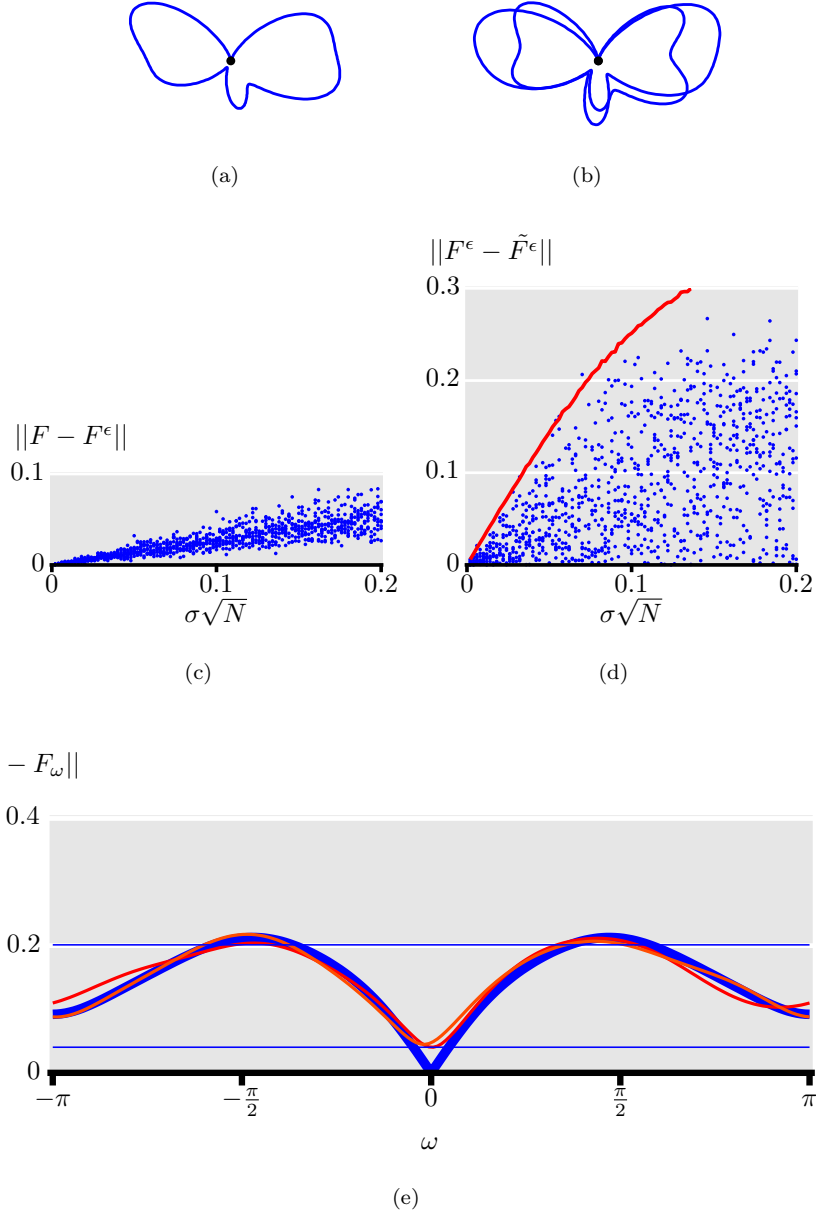


Figure 5: (a) A distribution, f . (b) Two of its noisy versions, f^ϵ . (c) The distance between f and its noisy versions plotted as a function of increasing noise. (d) The additional error introduced by F_1 canonicalization of the noisy versions, along with the expectation from Theorem 1. Because this distribution has a small $|F_1|$ value compared to the noise, F_1 canonicalization may make large changes to noisy versions of f . (e) Curves indicate $\|F - F_\omega\|$ (bold line) and $\|F - F_\phi^\epsilon\|$ (thin lines) for f , two noisy versions of f , and their rotations by ϕ . Because $|F_1|$ is small compared to the noise, $\|F - \tilde{F}_\phi^\epsilon\|$ (horizontal lines) is sometimes large.

Finding this distance requires an optimization over ϕ , so is not appropriate when many pairwise distances must be computed. A reasonable approximation, however, is

$$\|f - g\|_{F_k \text{ canonical}} = \min_{1 \leq \ell \leq K} \|\tilde{f}^\ell - \tilde{g}^\ell\|, \quad (10)$$

which only requires the calculation of K distances.

5. Experiments and Discussion

We now present experiments in keypoint description, person detection, and texture segmentation that show the promise of FS-KDEs using the \cos^{2K} kernel as a tool in image processing.

5.1. Keypoint Description

A typical approach to image registration involves selecting keypoints from the images to be matched, finding pairs of corresponding keypoints, and solving for the transform based on the location of these pairs. One way to find correspondences between keypoints is to calculate a keypoint descriptor from the pixels around each keypoint. When two keypoints correspond, the distance between their descriptors should be low; when they do not, it should be high. A good keypoint descriptor should be highly discriminative while simultaneously being invariant to the transform that the registration aims to reverse.

We evaluate the performance of the \cos^{2K} kernel as a keypoint descriptor using the University of British Columbia Multi-view Stereo Correspondence Dataset [20]. This dataset was constructed by extracting image patches around difference of Gaussian interest points in many images of the same few scenes (Statue of Liberty, Yosemite National Park, and Notre-Dame Cathedral). Depth maps of the scenes were used to determine which interest points match in 3D space, resulting in lists of corresponding and non-corresponding image patches (see Figure 6). The patches are 64×64 greyscale images; in our experiments we crop them to a circular region with a diameter of 60 pixels to avoid artifacts when rotating the patches.

We compare three simple keypoint descriptors: (i) The *raw intensity* descriptor is formed by concatenating the pixel values of the patch into a vector. It has dimension equal to the number of pixels in the patch, 2,828. (ii) The *gradient histogram* descriptor is formed by computing the image gradient at each pixel in the patch and forming a histogram of the gradient angles weighted by the norm of the gradient. The dimension of the descriptor is equal to the number of histogram bins; we vary it between 4 and 32. (iii) The \cos^{2K} descriptor is formed by computing the image gradient at each pixel and computing a \cos^{2K} FS-KDE of the gradient angles weighted by the norm of the gradient. We truncate the descriptors according to (8) with $\epsilon = 1 \times 10^{-5}$ and we vary the descriptor length between 4 and 32.

Additionally, we compare three canonical versions of these descriptors: (i) The *canonical gradient histogram*, which is the same as the gradient histogram but

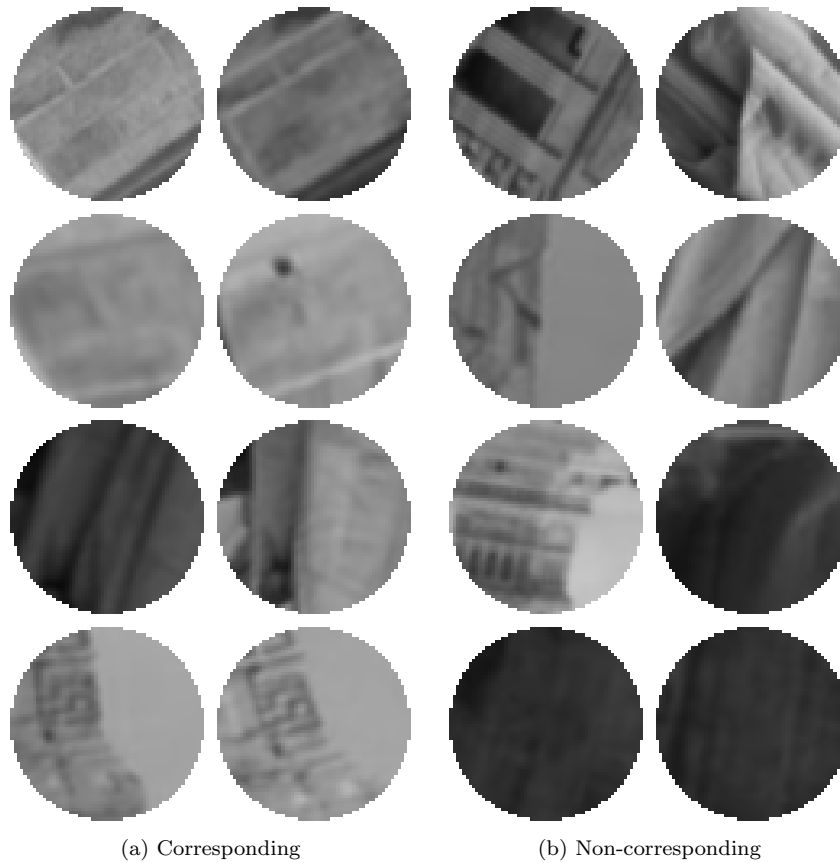


Figure 6: Examples of corresponding (left) and non-corresponding (right) pairs of patches from the British Columbia Multi-view Stereo Correspondence Dataset [20]. Though the patches are rotated to a canonical orientation, corresponding patches still exhibit viewpoint and intensity variation.

with its bins rotated so that the first bin has the largest value. (ii) The F_1 *canonical* \cos^{2K} , which follows the canonicalization procedure from (9). (iii) The F_k *canonical* \cos^{2K} , which follows the canonicalization procedure from (10).

We use each of these methods to compute descriptors for 250,000 pairs of corresponding patches and 250,000 pairs of non-corresponding patches. We then calculate the Euclidean distance between each pair of descriptors; good descriptors should assign small distances to corresponding patches and large distances to non-corresponding patches. Setting a threshold on this distance allows the descriptor to classify pairs of patches as corresponding or not. To quantify the performance of each descriptor, we calculate the area under its ROC curve (the curve formed when plotting true positive rate versus false positive rate over the whole range of possible threshold values). A perfect classifier has an area under ROC (AUC) of 1, while a random classifier has an AUC of .5.

Figure 7 shows the results of our comparison of keypoint descriptors in terms of AUC. The \cos^{2K} descriptor has the highest AUC for a wide range of descriptor sizes (6 to 26) and has the highest overall AUC of .83 at descriptor length 10). After this peak at descriptor length ten, its performance declines as the descriptor length increases, which is consistent with the idea that the descriptor becomes overly specific when it is long, increasing distances between corresponding patches. The error chance of the gradient histogram descriptor also increases with descriptor size, for the same reason. We attribute the gap in performance between the \cos^{2K} and gradient histogram descriptors to the \cos^{2K} kernel’s smooth handling of small rotations: even though the patches in the dataset are rotated to a canonical orientation, small rotations do exist between corresponding patches, which could distort the gradient histograms (see Figure 6 for examples). The canonical versions of the \cos^{2K} and histogram descriptors perform generally worse than their non-canonical counterparts, which is as expected since no canonicalization should be necessary for this dataset. In this case, canonicalization will decrease the distance between non-corresponding patches more than it does for corresponding patches, increasing the number of false positives at a given threshold.

In a separate experiment, we randomly rotated each patch in the dataset and ran the same comparison (Figure 8). As expected, this greatly increased the error rate for the histogram and \cos^{2K} descriptors, as they are not rotation invariant without canonicalization. The canonical versions of these descriptors were mostly unaffected by the change, since they are rotation invariant descriptors. Both canonical versions of the \cos^{2K} descriptor were superior to the canonical gradient histogram, which we hypothesize is due to the robustness to noise of the proposed canonicalizations. The F_k canonicalization was better than the F_1 canonicalization, which suggests that symmetries of the type discussed in Section 4, which break the F_1 canonicalization, do exist in this dataset.

5.2. Person Detection

We now evaluate the usefulness of the FS-KDE using the \cos^{2K} kernel as a feature in a person detection application. A typical approach to person detec-

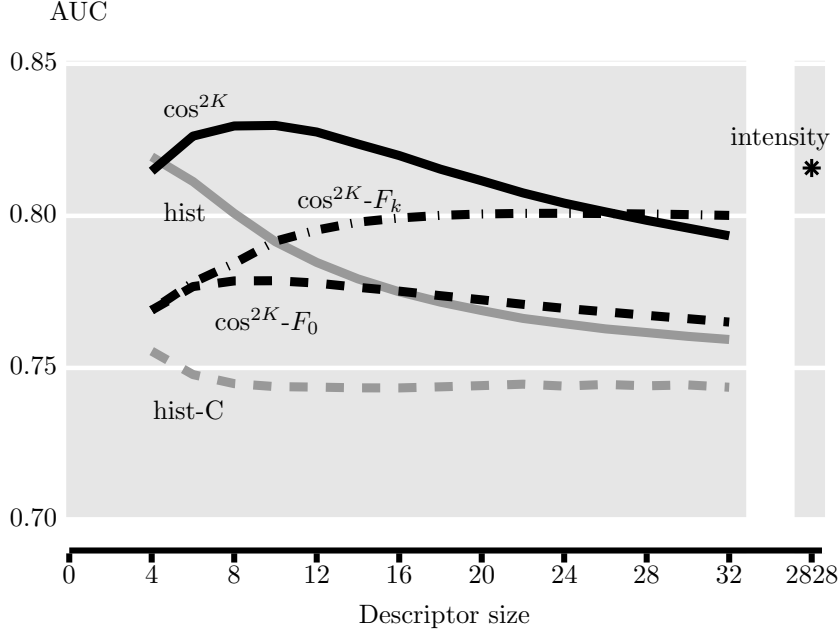


Figure 7: AUC for intensity, gradient histogram, and the proposed \cos^{2K} descriptors of varying size. The best performance is achieved by the \cos^{2K} descriptor of size ten. The canonical descriptors perform poorly in this experiment because the patches are already in a canonical orientation.

tion (or, in general, object detection), is to train a classifier on features which consist of distributions of angles. To preserve some spatial information, these distributions are calculated in a few windows of the input, e.g., upper left, upper right, lower left, lower right, and then concatenated together to form the final feature vector.

We use the INRIA person dataset [10] to do a comparison of feature detectors for human detection. This dataset is intended for supervised classification of images as containing a person (positive) and not containing a person (negative). It includes a training set of 2,416 positive and 1,218 negative images and a testing set of 1,126 positive and 453 negative images. For all images, we use the center 64×128 pixels for feature extraction.

We compare the following feature extractors: (i) *Raw intensity* simply uses the pixels of the image as features and therefore has length 8192. (ii) *Gradient histogram* separates the image into 8×8 -pixel blocks and computes a histogram of gradients inside each block, concatenating these histograms into a feature vector. We vary the number of bins per block from 4 to 64. (iii) *HOG*, originally described in [10], also forms gradient histograms from blocks of the input image, but includes an additional block normalization step that can increase the feature’s illumination invariance. We use the implementation in [21] and vary

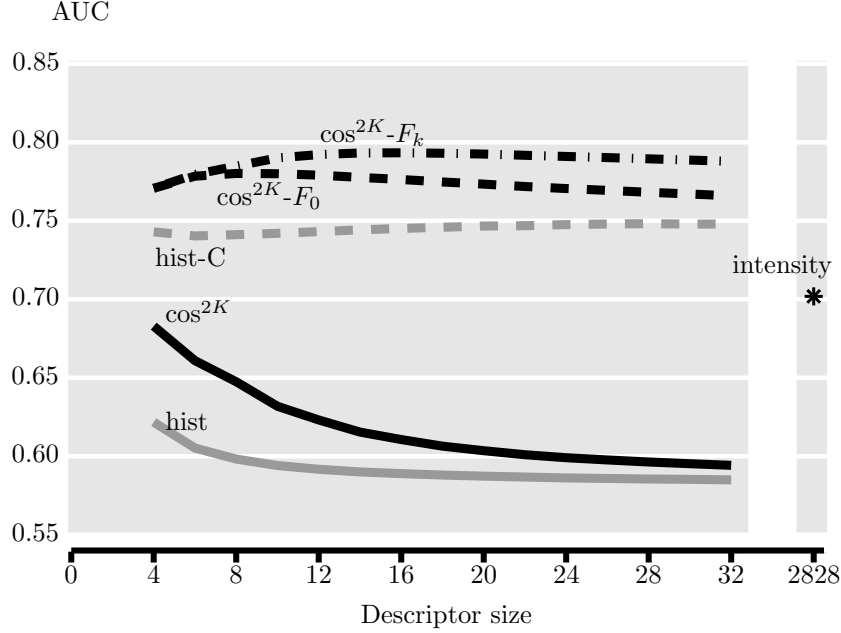


Figure 8: AUC for intensity, gradient histogram, and the proposed \cos^{2K} descriptors of varying size on patches with random orientation. The \cos^{2K} FS-KDE and gradient histogram descriptors cannot handle patch rotations, so have much higher error chance here than in Figure 7. The best performance is now achieved by the \cos^{2K} FS-KDE descriptor with F_k canonicalization.

the number of orientations per block from four to sixty-four. (iv) \cos^{2K} forms truncated ($\epsilon = 1 \times 10^{-5}$) \cos^{2K} FS-KDEs for 8×8 -pixel blocks of the input. We vary the descriptor length per block from four to sixty-four.

We use each of these methods to extract features from the training and test sets. For each set of features, we train a linear SVM classifier (in MATLAB) on the training set, then use it to classify each image in the test set as negative or positive. In a separate experiment, we create new testing and training datasets by rotating each image between 15 and -15 degrees uniformly at random.³ This rotation is not enough that canonicalization is necessary, but is meant to test the robustness of the features to small rotations. We report accuracy, the number of correctly classified images in the testing set divided by the total number of images in the testing set, from both experiments in Figure 9.

For the unrotated set, the HOG features have the highest accuracy, except when the feature vector size is very small (four numbers per image block, resulting in 512 numbers per image). We suspect HOG’s increase in performance over

³This rotation does not produce edge artifacts because we crop the central portion of a larger image to form the training and testing images.

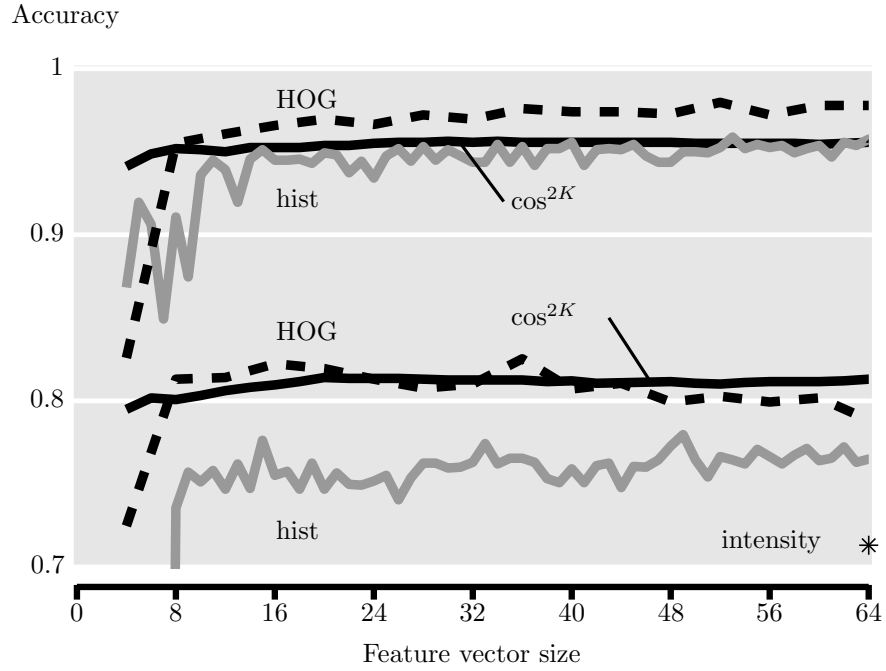


Figure 9: Accuracy on a person detection task for the intensity, gradient histogram, HOG, and \cos^{2K} feature extractors, plotted as a function of the feature vector length per image block. The top set of lines is for the INRIA person dataset [10], the bottom set is for the same dataset with small random rotations added. Without rotations, HOG features have the highest accuracy for most lengths and the intensity features (as expected) have the lowest. With rotations, the performance of all four methods declines, but the \cos^{2K} FS-KDE declines the least, leaving it with accuracy comparable to HOG. The accuracy of the intensity features for the rotated dataset was below chance and not plotted.

the gradient histogram and KDE features comes from the normalization scheme used in HOG, which gives it an invariance to illumination changes missing in the other methods. The low accuracy of the intensity features is as expected, given that greyscale intensity is not a reliable way to distinguish people from background clutter. The accuracy of the histogram gradient and both KDEs features are similar, except that the accuracy of the histogram features is less stable as the feature vector size changes. We attribute this to the binning effects introduced by the histogram. We also note that the decline in performance as descriptor length increases seen in Figure 7 is not evident here because we use an SVM as opposed to simply calculating distances.

When a small amount (± 15 degrees) of rotation is added to the images in the dataset, the accuracy of all the feature sets decreases, but the decrease is smallest for the \cos^{2K} features. The rotation makes intensity features worse than chance (not plotted in Figure 9) because these features have no invariance to rotation. We suspect that binning artifacts (as discussed in Figure 1) explain the relatively larger decrease in accuracy for the gradient histogram and HOG features, because they both rely on gradient histograms. This experiment shows that the smooth \cos^{2K} kernel provides greater invariance to small rotations than the binning employed by histograms, resulting in higher accuracy in the person detection task.

5.3. Texture Segmentation

Distributions of angles are also useful as texture features. In our previous work, [22], we presented an algorithm for segmentation based on unmixing the local color histograms of an input image, which we call the Occlusion of Random Textures SEGmenter, (ORTSEG). In this experiment, we extend ORTSEG to include distributions of angles as well. We compare the effectiveness of histograms versus \cos^{2K} FS-KDEs to capture these distributions of angles.

We compare the methods on the *random texture dataset* from [22] plus an additional synthetic dataset, which we refer to as *dead leaves*. The images in the random texture dataset set each comprise three textures with different color distributions and no meaningful edge information (see [22] for more details and examples). The dead leaves dataset (Figure 10) images each comprise three textures with the same color distributions but differently oriented edges. To create this dataset, we first pick three seed locations at random and use them to partition the image into the three Voronoi regions. We then generate the image via a dead leaves procedure: we sequentially place shapes of random color into the image at random locations until every pixel is covered. Depending on which of the three regions a shape lands in, it is selected to be either a vertical bar, horizontal bar, or diagonal bar. We select the ground truth label of each pixel to correspond to the shape that covered it most recently.

We compare three methods on this dataset. (i) *ORTSEG* is the original segmentation system described in [22], which relies only on color histograms. (ii) *ORTSEG-hist* uses both color histograms and local gradient histograms, with the number of bins selected from training between eight and 40. (iii) *ORTSEG-FS-KDE* uses color histograms and local \cos^{2K} FS-KDEs of gradients, with no

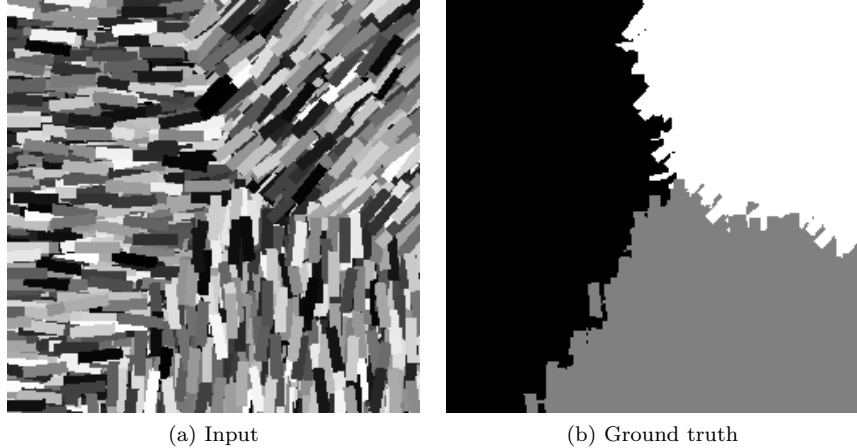


Figure 10: An example image and corresponding ground truth from the dead leaves dataset. For images like these, angular distributions are an important feature.

canonicalization, and with the number of complex coefficients selected from training between four and 20. We do not evaluate canonical versions of these methods because canonicalization will make the angular distributions in the different texture regions of the dead leaves images match, resulting in low segmentation accuracy. The choice of whether or not to canonicalize for segmentation depends on whether textures that match except for their orientation are intended to be grouped together or not.

The experiment is structured exactly as in [22]; in short, it is a leave-one-out cross validation. The results are reported in terms of Rand index [23, 24], which measures the fraction of pairs of pixels that are either in the same region in both the segmentation result and ground truth or in different regions in both the segmentation result and ground truth. It therefore ranges between zero and one, with one being perfect agreement with the ground truth.

The results of the segmentation experiment are given in Table 1. The three methods perform equally well on the random texture dataset, which makes sense because color information alone is enough to distinguish the textures. On the dead leaves dataset, the basic ORTSEG method, which relies only on color, cannot distinguish the textures at all and thus performs poorly. The gradient histogram and FS-KDE versions of ORTSEG improve performance by including edge information. That performance increase is most pronounced for the FS-KDE version. We attribute this difference to the smooth \cos^{2K} kernel used in the FS-KDE giving better robustness to small variations in gradient angle as compared to histograms. These results serve as a proof of concept for the efficacy of the \cos^{2K} FS-KDE for including gradient information into the segmentation method ORTSEG.

Table 1: Comparison of the basic ORTSEG method with versions using gradient histograms and \cos^{2K} FS-KDEs. The augmented versions improve the performance on the dead leaves dataset, where edge information is critical.

Method	Dataset	
	random texture	dead leaves
basic	0.989 \pm 0.002	0.551 \pm 0.087
hist	0.988 \pm 0.002	0.702 \pm 0.161
FS-KDE	0.989 \pm 0.002	0.907 \pm 0.094

6. Conclusion

In this work, we presented a new bandlimited Gaussian-like kernel, useful for describing angular distributions in computer vision applications. Because the kernel is bandlimited, the resulting KDEs are also bandlimited and therefore can be represented exactly by a finite number of their Fourier series coefficients, a technique which we call FS-KDE. Though this type of density estimation is not new, it has not been much explored in image processing, where finite-length angular descriptors are very useful. We also presented a canonicalization scheme for FS-KDEs which allows them to create rotation invariant descriptions of angular distributions and analyzed the robustness of this scheme to noise.

In our experiments, we compared FS-KDEs using our proposed kernel to histograms in the contexts of patch matching, person detection, and texture segmentation. In the patch matching experiment, the FS-KDE descriptors outperformed histogram-based descriptors both when patches were upright and when they were randomly oriented. The person detection experiment showed that FS-KDE features provide higher person detection accuracy than histogram features, especially when a small amount of random rotation was added to the dataset. Finally, the segmentation experiment suggested that the FS-KDE is a better way to capture texture information than histograms in the context of texture segmentation. Taken together, these experiments provide strong proof of concept for the efficacy of FS-KDEs using our new bandlimited kernel as tools for describing distributions of angles in image processing applications.

7. Acknowledgements

The authors gratefully acknowledge support from the NSF through awards 0946825 and 1017278, the Achievement Rewards for College Scientists Foundation Scholarship, the John and Claire Bertucci Graduate Fellowship, the Philip and Marsha Dowd Teaching Fellowship, and the CMU Carnegie Institute of Technology Infrastructure Award.

Appendix A. Proof of Theorem 1

PROOF.

$$\begin{aligned}
\|F^\epsilon - \tilde{F}^\epsilon\| &\stackrel{(a)}{=} \sqrt{\sum_{k=-K}^K |F_k^\epsilon - e^{-jk \arg(F_1^\epsilon)} F_k^\epsilon|^2} \\
&\stackrel{(b)}{=} \sqrt{\sum_{k=-K}^K (|F_k^\epsilon| |1 - e^{-jk \arg(F_1^\epsilon)}|)^2} \\
&\stackrel{(c)}{=} \sqrt{\sum_{k=-K}^K \left(|F_k^\epsilon| 2 \sin\left(\frac{k}{2} \arg(F_1^\epsilon)\right) \right)^2}, \tag{A.1}
\end{aligned}$$

where (a) follows from the definition of norm, (b) from factoring, and (c) from Euler's formula and the fact that $|e^{j\theta}x| = |x|$ for all x .

In order to find $\arg(F_1^\epsilon)$ in (A.1), we note that $\arg(F_1^\epsilon) = \arg(F_1^\epsilon/\alpha)$ and that

$$F_1^\epsilon/\alpha = B_1 \sum_{n=0}^{N-1} w_n e^{-j\theta_n} + \epsilon_n = F_1 + B_1 \sum_{n=0}^{N-1} \epsilon_n$$

by assumption and the definition of the FS-KDE (1). Therefore

$$\begin{aligned}
\arg(F_1^\epsilon) &= \arctan\left(\frac{\Im(F_1^\epsilon/\alpha)}{\Re(F_1^\epsilon/\alpha)}\right) \\
&= \arctan\left(\frac{B_1 \sum_{n=0}^{N-1} \Im(\epsilon_n)}{|F_1| + B_1 \sum_{n=0}^{N-1} \Re(\epsilon_n)}\right), \tag{A.2}
\end{aligned}$$

where $\Re(z)$ and \Im denote the real and imaginary parts of z , respectively, because F_1 is real by assumption.

To bound $|F_k^\epsilon|$, we use the fact that $|B_k \sum_{n=0}^{N-1} w_n e^{-jk\theta_n}| \leq |B_k \sum_{n=0}^{N-1} w_n|$ for any choice of (Θ, W) , meaning that

$$|F_k^\epsilon| \leq B_k N.$$

To finish the proof, we replace the sums in (A.2) with new random variables ϵ and v , the distributions of which we know because of the noise model assumed in the proof.

References

- [1] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Kerkyra, Greece, 1999, pp. 1150–1157 vol.2.
- [2] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

- [3] E. Mortensen, H. Deng, and L. Shapiro, “A SIFT descriptor with global context,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 1, San Diego, CA, Jun. 2005, pp. 184–190 vol. 1.
- [4] A. Abdel-Hakim and A. Farag, “CSIFT: A SIFT descriptor with color invariant characteristics,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 2, New York, NY, 2006, pp. 1978–1983.
- [5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Comput. Vis. Image. Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [6] Y. Ke and R. Sukthankar, “PCA-SIFT: a more distinctive representation for local image descriptors,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 2, Washington, DC, Jun. 2004, pp. II-506 – II-513 Vol.2.
- [7] A. Sotiras, C. Davatzikos, and N. Paragios, “Deformable medical image registration: A survey,” *IEEE Trans. Med. Imag.*, vol. 32, no. 7, pp. 1153–1190, Jul. 2013.
- [8] J. Aggarwal and M. Ryoo, “Human activity analysis: A review,” *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011.
- [9] A. Ramanan and M. Niranjan, “A review of codebook models in patch-based visual object recognition,” *J. Signal Process. Syst.*, vol. 68, no. 3, pp. 333–352, Sep. 2012.
- [10] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, vol. 1, 2005, pp. 886–893.
- [11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
- [12] D. Geronimo, A. Lopez, A. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239–1258, Jul. 2010.
- [13] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
- [14] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, “Fast computation of rotation-invariant image features by an approximate radial gradient transform,” *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 2970–2982, Aug. 2013.
- [15] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. New York: Springer, Sep. 2004.

- [16] A. Feuerverger and R. A. Mureika, “The Empirical Characteristic Function and Its Applications,” *Ann. Stat.*, vol. 5, no. 1, pp. 88–97, Jan. 1977.
- [17] K. Liu, H. Skibbe, T. Schmidt, T. Blein, K. Palme, T. Brox, and O. Ronneberger, “Rotation-Invariant HOG Descriptors Using Fourier Analysis in Polar and Spherical Coordinates,” *Int. J. Comput. Vis.*, vol. 106, no. 3, pp. 342–364, Jun. 2013.
- [18] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge: Cambridge University Press, 2014, <http://www.fourierandwavelets.org/>. [Online]. Available: <http://www.fourierandwavelets.org/>
- [19] M. T. McCann, M. Fickus, and J. Kovačević. (2015) Fourier series of kernel density estimates for rotation invariant angular distribution estimation. [Online]. Available: http://www.jelena.ece.cmu.edu/repository/rr/15_McCannFK/15_McCannFK.html
- [20] S. Winder, G. Hua, and M. Brown, “Picking the best DAISY,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, Miami Beach, FL, Jun. 2009, pp. 178–185.
- [21] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms,” <http://www.vlfeat.org/>, 2008.
- [22] M. T. McCann, D. G. Mixon, M. C. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovačević, “Images as occlusions of textures: A framework for segmentation,” *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2033–2046, May 2014.
- [23] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *J. Am. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971. [Online]. Available: <http://dx.doi.org/10.1080/01621459.1971.10482356>
- [24] R. Unnikrishnan, C. Pantofaru, and M. H. Hebert, “Toward objective evaluation of image segmentation algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007. [Online]. Available: <http://dx.doi.org/10.1109/tpami.2007.1046>